

## **AMENDMENTS TO THE CLAIMS**

1-41. (Cancelled)

42. (Currently Amended) A method comprising:

executing a core module;

initializing a plurality of firmware modules by the core module, wherein the initializing comprises:

examining at least two firmware modules to determine a required order of dispatch of the firmware modules;

dispatching an earlier of the two firmware modules by the core module and then dispatching a later of the two firmware modules by the core module;

after initializing the plurality of firmware modules, initializing a system memory; and

invoking a function contained in a third firmware module by the earlier of the two firmware modules during a dispatch of the earlier firmware module or by the later of the two firmware modules during a dispatch of the later firmware module.

43. (Previously Presented) The method of claim 42 wherein examining comprises checking a resource list of each of the two firmware modules.

44. (Currently Amended) The method of claim 42 wherein the initializing of the plurality of firmware modules further comprises:

~~executing a core module;~~

determining the requirements configuration of the plurality of firmware modules appropriate to run; and

examining a platform to determine whether hardware to be initialized by the module is present in the platform.

45. (Previously Presented) The method of claim 42, further comprising:

marking a data structure after dispatching a firmware module, wherein the data structure is to indicate whether the firmware module has been dispatched.

46. (Previously Presented) The method of claim 45 wherein the data structure is a bit array.

47. (Previously Presented) The method of claim 45 wherein the data structure is held in a processor register.

48. (Currently Amended) The method of claim 42,  
wherein the third firmware module is dispatched before the function contained in the third firmware module is invoked.

49. (Previously Presented) The method of claim 48 wherein invoking comprises:  
scanning a daisy chain of firmware modules to find a module that operates with a hardware component present in a platform; and  
invoking the function if such a module is found.

50. (Currently Amended) A computer-readable medium containing instructions to cause a programmable processor to perform operations comprising:

determining dependencies among a plurality of firmware modules by a separate core module based on information about services imported and exported by each of the firmware modules before dispatching the plurality of firmware modules by the separate core module, and the services imported by each firmware module of the plurality of firmware modules dispatched by the separate core module are provided in another firmware of the plurality of firmware modules dispatched by the separate core module;

dispatching each module of the plurality of firmware modules in an order that satisfies the dependencies; and

initializing a volatile system memory after dispatching the plurality of firmware modules.

51. (Previously Presented) The computer-readable medium of claim 50 wherein the information about services imported and exported by a firmware module comprises:  
an import table containing at least one service that may be called by the firmware module;  
or  
an export table containing at least one service provided by the firmware module; or  
the import table and the export table.
52. (Previously Presented) The computer-readable medium of claim 50, containing additional instructions to cause the programmable processor to perform operations comprising:  
maintaining a data structure to indicate which firmware modules of the plurality of firmware modules have been dispatched, wherein the data structure is kept in a register of the programmable processor.
53. (Previously Presented) The computer-readable medium of claim 50 wherein dispatching a module comprises saving a return address in a processor register and executing a beginning instruction of the module.
54. (Currently Amended) A system comprising:  
a hardware component to perform a function;  
a volatile memory that can store data after the volatile memory is initialized; and  
a non-volatile memory containing a Basic Input-Output System (“BIOS”) including a BIOS core and a plurality of firmware modules,  
wherein two of the plurality of firmware modules are to initialize hardware components to perform the function, and  
one of the two firmware modules is to initialize a hardware component that is not present in the system by iterating through each of the plurality of firmware modules within the BIOS to determine whether the hardware component is present, and  
two of the plurality of firmware modules are to be dispatched before the volatile memory is initialized.

55. (Previously Presented) The system of claim 54 wherein the volatile memory is a Random Access Memory (“RAM”).

56. (Previously Presented) The system of claim 54 wherein the non-volatile memory is at least one of a Read-Only Memory (“ROM”) or a Flash memory.

57. (Cancelled)